

## תרגיל מס' 1: חיפוש במבוך – מימוש אלגוריתם ID

מצורפים הקבצים הבאים:

`stack.py` – מימוש מבנה הנתונים: מחסנית.

`frontier.py` – מימוש מבנה נתונים המתאים ל-ID: מחסנית "משוכללת" – מחסנית שמאחסנת

מצבים עד עומק חיפוש `d`; כאשר היא מתרוקנת היא טוענת שוב את המצב ההתחלתי ומגדילה את `d` ב-1.

`state.py` – מימוש מחלקה של `Maze` בגודל `nXn`.

`Search.py` – מימוש של החיפוש, כפי שנלמד בהרצאה.

עליכם לעבור על הקבצים המצורפים, להבין אותם ולבצע את המשימות הבאות:

1. שנו את מבנה הנתונים כך שישמור את מספר המצבים המאוחסנים כרגע במבנה הנתונים.

כתוצאה מזה, `frontier` יכלול 5 פריטים ולא 4:

[stack, max. depth, initial. state, try next level?, total items pushed]

2. צרו `frontierStack` שיממש מחסנית, וישמור את הפרטים הבאים:

[Stack, total item pushed]

3. הגדירו מחלקה `Queue` שתממש התנהגות של תור תוך שימוש ברשימה.

צרו `frontierQueue` שיממש תור, וישמור את הפרטים הבאים:

[Queue, total item pushed]

4. כתבו תוכנה שתריץ 100 הרצות של בניית מבוכים בגודל `5*5` ומציאת פתרון עבורם, בשיטת `DFS`,

`BFS`, ו-`Iterative Deepening` ותדפיס טבלה עם המידע הבא:

ממוצע בדיקות שנדרשו עבור מבוכים שנתקעו (שלא היה להם פתרון), עד שהאלגוריתם עצר והודיע שאין פתרון	אורך מקסימלי של מספר המצבים שהיו בתוך התור בזמנית (ממוצע עבור המבוכים שהצליחו להגיע אל היעד)	מספר בדיקות ממוצע (ממוצע עבור המבוכים שהצליחו להגיע ליעד).	אחוז המבוכים שנמצא בהם מסלול מהמקור ליעד	
				DFS
				BFS
				Iterative Deepening

5. מה מסקנתכם?

בהצלחה רבה!